Deep-LfD: deep robot learning from demonstrations

Amir Ghalamzan E.[†] and Kiyanoush Nazari

[†] *corresponding author; University of Lincoln, Lincoln LN6 7TS, United Kingdom;*

Abstract

Like other robot learning from demonstration (LfD) approaches, deep-LfD builds a task model from sample demonstrations. However, unlike conventional LfD, the deep-LfD model learns the relation between high dimensional visual sensory information and robot trajectory/path. This paper presents a dataset of successful needle insertion by da Vinci Research Kit into deformable objects based on which several deep-LfD models are built as a benchmark of models learning robot controller for the needle insertion task.

Keywords: deep learning, robot learning from demonstration, deformable object manipulation

1. Context and Motivation

Programming robots are time-consuming and costly [1]. Conventional LfD methods were proposed to reduce the programming cost and time. However, the conventional LfD are not sufficient for many complex tasks, e.g. robotic surgery. Minimally Invasive Surgery (MIS) [2, 3] has been improved by surgical robots. Robotic suturing is key to every MIS that takes considerable time and effort during every operation. Our proposed approach builds a baseline of a deep-LfD that contributes to a future fully autonomous robotic suturing in real surgery operations. This will significantly reduce the operation time, cost and fatigue and cognitive workload on the surgeon operators, hence, better patients experiences.

One of the challenges of successful completion of robotic suturing is tissue deformation while inserting the needle in the tissue that makes the stitch fail since it does not grip enough tissue [4]. Fig 1(a) shows a circular needle being inserted into soft tissues to achieve the desired exit point (red sphere); but, the actual exit point is different from the desired one as the needle deforms the tissue. Hence, the second arm is usually used to actively manipulate the tissue to ensure the desired and actual exit points are the same.

Although conventional motion planning and control are proposed for programming robotic tasks [5, 6, 7, 8, 9], they are not generalisable across different tasks, as they need hand-designed and task-specific features. On the contrary, we propose deep robot learning from demonstration (Deep-LfD) as a solution to motion planning and controller in such an unstructured environment that extract features from high dimensional observation space, namely RGB images.

2. Software details and impact

As a benchmark for Deep-LfD of the robot controller, our software on `https://github.com/imanlab/d-lfd` includes a comprehensive needle insertion dataset [10], feedforward neural networks (NN) and Recurrent NN models. The software includes (i) Preprocessing scripts, (ii) feedforward models scripts, and (iii) recurrent convolutional networks (RCNs) scripts [11]. Preprocessing module turns the videos of the camera view into image frames. Preprocessing module keeps the sampling frequency of the image frames and the robot data the same. Hence, the number of images for different needle insertion trails varies as their durations are different and functions of the insertion and desired exit points. Furthermore, we incorporated these models in a closed-loop control system (deep-LfD) that observes the word state (namely needle insertion in soft tissue) via RGB cameras calibrated w.r.t. the robot base frame. It then generates the control action for time t at time t yielding the desired state at $t + 1$. Our software is based on two major python libraries (1) Tensorflow–a library from Google developed for deep learning, and (2) Keras– a library built on top of Tensorflow as a high-level API for deep neural networks design. This software provides a baseline for further development of Deep-robot learning from demonstrations.

2.1. Dataset

We have created a dataset of inserting a needle into a deformable object using a da Vinci Research Kit [10, 11]. Fig. 2 shows the experimental setup used for data collection. The phantoms (deformable objects used in our experiments) are made of homogeneous polyethylene whose Young's modulus is circa 0.02-0.04 [GPa]. In robotic surgery, Arm-2 is used to manipulate the soft tissue (Fig.1(c)) to ensure the needle passes/exits through the desired point yielding enough gripped tissue by the stitch. Fig.2(b) presents the structure of the dataset included for each needle insertion trial, which consists of (i) 60 successful needle insertion trials with randomised desired exit points recorded by 6 high-resolution calibrated cameras, (ii) the corresponding robot's data, calibration parameters and (iii) the commanded robot's control input where all the collected data are *synchronised*.

2.2. Deep-LfD Models

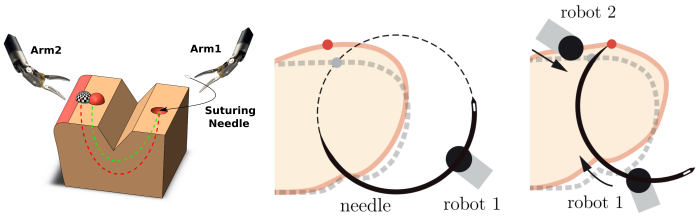
Our baseline model is our custom-designed CNN for feature extraction, we call it FNN. We compared the performance of FNN with those of various well-known CNN models including AlexNet, VGG16, VGG19, MobileNet, and ResNet152V2. In the feedforward models, after the CNN, dense layers are used to generate the control action at t, yielding the robot state at $t+1$. Table 2.3 shows FNN error of state-of-the-art models and FNN.

To further improve the accuracy of the control action generation, we use recurrent neural networks (RNNs)–including simple RNN, Gated Recurrent Unit (GRU), and Long Short Term Memory (LSTM) are used– that takes the low dimensional feature vectors outputted by feedforward models and output the control action at time t. This (using RNNs instead of the dense layers in the simple feedforward models) significantly improves the accuracy of the generated control actions. Fig.3 presents the overall structure of the deep-LfD used for robot control action generation.

2.3. Software impact

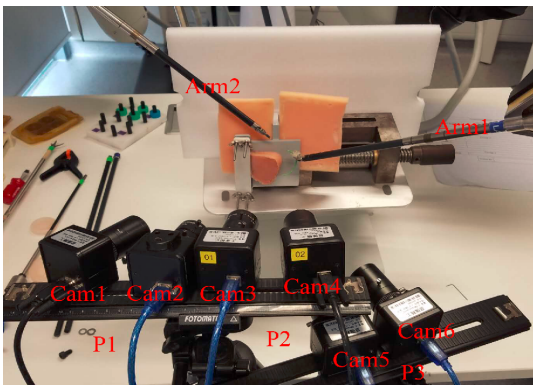
This software enables deep-LfD that will significantly impact most of the tedious teleoperation tasks, e.g. robotic surgery and teleoperate systems in extreme/remote environments– such as space robot teleoperation. Deep-LfD is a baseline implementation of a data-driven robot controller that is an end-to-end solution for learning robotic tasks requiring motion planning and control. The software is a baseline for data-driven learning from demonstrations which is the first of its type. There are several open-research questions, hence, the dataset and baseline implementations provide a benchmark for future research development. The dataset is unique having several interesting features that enable future research, e.g. (1) generalising robot controller to different camera views, (2) closing the control loop via data-driven approach and RGB cameras; (3) data-driven receding horizon motion planning for deformable object manipulation; (4) data-driven robot controller. This software provides a novel means of looking at robot programming to perform complex tasks where the design of classical control systems are time-consuming and expensive (nonetheless, the generation of a corresponding dataset is doable–similar to teleoperated tasks in robotic). This software can learn the controller of tedious teleoperated robotic tasks, e.g. robotic sort-and-segregation and robotic surgery where it is hard to model the dynamic of robot-environment interaction. Works in [10, 11] are based on the software presented in this paper. Moreover, we are further developing the software to use it for (1) soft fruit pushing [12]: in which the model learns how to reach a ripe fruit and pick it from sample successful fruit cluster manipulation ; and (2) slippage avoidance during manipulative movements [13]: in which the demonstrations

successfully controlling the robot (collected via teleoperation or Kinesthetic teaching) are used to learn a manipulation control model. Before using this software for controlling unstable or marginally stable systems, further testing and stability analysis are required [14]. Reproducible software capsule can be found at `https://codeocean.com/capsule/0078633/tree/v1`

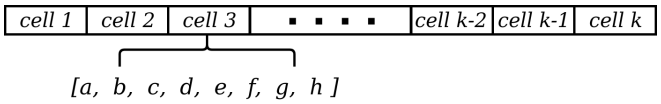


(a) (b) (c)

Figure 1: Needle insertion in soft tissue: (a) the needle pushes the tissue making the actual (red sphere) and desired (black-white sphere) needle exit points as well as the actual (green dashed line) and desired (red dashed line) needle paths different; (b) A 2D schematic of tissue deformation due to needle insertion (original shape shown as grey dash line) alter the target (red dot) from its original position (grey dot); and (c) Arm 2 is actively manipulating the tissue to improve insertion accuracy, as adopted by manual suturing skills.



(a)



(b)

Figure 2: (a) Dataset collection setup; (b) dataset includes a : joint space kinematic data of Arm 1 (1×6); b: joint space kinematic data of Arm 2 (1×6); c : pose of Arm 1 w.r.t its base frame (4×4); d : pose of Arm 2 w.r.t its base frame (4×4); e: pose of Arm 1 w.r.t its base frame at $t+1$; f: pose of Arm 2 w.r.t its base frame at $t+1$; g: 2D tracking target point on the image captured by 6 cameras (6×2); h: computed 3D position of the target point w.r.t Camera 3 (1×3)

Acknowledgements

This work was partially supported by Centre for Doctoral Training (CDT) in Agri-Food Robotics (AgriFoRwArdS) Grant reference: EP/S023917/1; and Research England fund to create Lincoln Agri Robotics (LAR) Centre

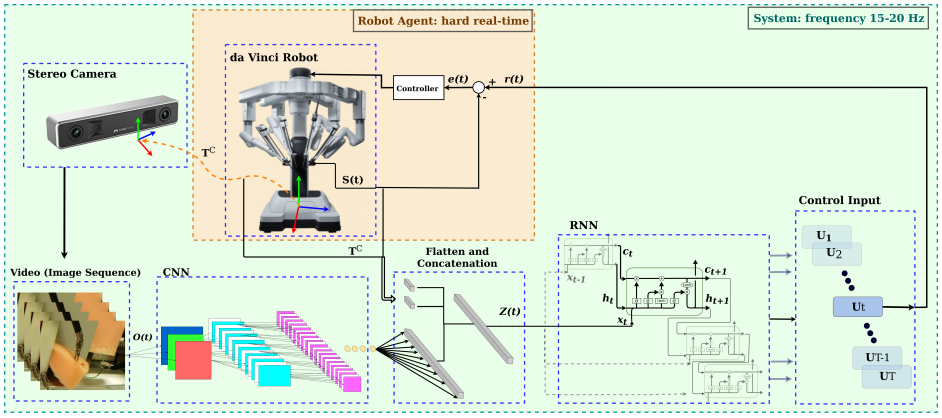


Figure 3: System demonstration: Robot agent, Camera, and the network including CNN, concatenation and RNN blocks.

Table 1: Evaluation of RCNs including fully general RNN, GRU and LSTM.

Model	MaxPE†	AvgPE†	MaxOE†	AvgOE†	Loss
Feedforward	0.154	0.755	1.528	0.475	0.005
RNN	0.068	0.153	0.704	0.173	0.0097
GRU	0.029	0.188	0.370	0.119	0.0051
LSTM	0.035	0.134	0.338	0.106	0.0048

† PE in $[mm]$ and OE in *degree*
† e−4

Table 2: Evaluation of D-RLfD with CNN architectures

Model	MaxPE†	AvgPE‡	MaxOE	AvgOE	Loss
AlexNet	0.212	1.983	2.518	1.027	0.053
VGG19	0.166	1.65	1.641	0.267	0.054
MobileNet	0.173	0.825	1.077	0.389	0.049
ResNet	0.280	0.964	1.631	0.477	0.050
Our CNN	0.154	0.755	1.528	0.475	0.005

† PE in $[mm]$ and OE in *degree*
‡ e−3

of Excellence.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v1.0
C2	Permanent link to code/repository used for this code version	`https://github.com/imanlab/d-lfd`
C3	Permanent link to Reproducible Capsule	
C4	Legal Code License	MIT License
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Python3, Matlab
C7	Compilation requirements, operating environments & dependencies	Python3, Tensorflow, sklearn, Pandas, Numpy, CUDA(optional)
C8	If available Link to developer documentation/manual	github repository README
C9	Support email for questions	aghalamzanesfahani@lincoln.ac.uk

Table 3: Code metadata (mandatory)

[1] S. Schaal, Is imitation learning the route to humanoid robots?, Trends in cognitive sciences 3 (6) (1999) 233–242.

[2] A. Mirbagheri, F. Farahmand, S. Sarkar, A. Alamdar, M. Moradi, E. Afshari, The sina robotic telesurgery system, in: Handbook of Robotic and Image-Guided Surgery, Elsevier, 2020, pp. 107–121.

[3] D. J. Jacofsky, M. Allen, Robotics in arthroplasty: a comprehensive review, The Journal of arthroplasty 31 (10) (2016) 2353–2363.

[4] F. Zhong, Y. Wang, Z. Wang, Y.-H. Liu, Dual-arm robotic needle insertion with active tissue deformation for autonomous suturing, IEEE Robotics and Automation Letters 4 (3) (2019) 2669–2676.

[5] C. Ng, W. Liang, C. W. Gan, H. Y. Lim, K. K. Tan, Precision motion control using nonlinear contact force model in a surgical device, in: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2019, pp. 5378–5381.

[6] K. Hauser, R. Alterovitz, N. Chentanez, A. Okamura, K. Goldberg, Feedback control for steering needles through 3d deformable tissue using helical paths, Robotics science and systems: online proceedings (2009) 37.

[7] P. Sundaresan, B. Thananjeyan, J. Chiu, D. Fer, K. Goldberg, Automated extraction of surgical needles from tissue phantoms, in: 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), IEEE, 2019, pp. 170–177.

[8] R. Alterovitz, K. Goldberg, A. Okamura, Planning for steerable bevel-tip needle insertion through 2d soft tissue with obstacles, in: Proceedings of the 2005 IEEE international conference on robotics and automation, IEEE, 2005, pp. 1640–1645.

[9] K. B. Reed, A. Majewicz, V. Kallem, R. Alterovitz, K. Goldberg, N. J. Cowan, A. M. Okamura, Robot-assisted needle steering, IEEE robotics & automation magazine 18 (4) (2011) 35–46.

[10] H. Hashempour, K. Nazari, F. Zhong, et al., A data-set of piercing needle through deformable objects for deep learning from demonstrations, arXiv preprint arXiv:2012.02458.

[11] A. Ghalamzan-E, Learning needle insertion from sample task executions, arXiv preprint arXiv:2103.07938.

[12] S. Mghames, M. Hanheide, A. Ghalamzan Esfahani, et al., Interactive movement primitives: Planning to push occluding pieces for fruit picking, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[13] A. Cavallo, G. De Maria, C. Natale, S. Pirozzi, Slipping detection and avoidance based on kalman filter, Mechatronics 24 (5) (2014) 489–499.

[14] W. Li, A. Leonardis, M. Fritz, Visual stability prediction for robotic manipulation, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 2606–2613.